Regras

June 5, 2019

1 0. Introdução

Trabalho:
Aluno: Maicon Dall'Agnol
R.A.: 151161868
Disciplina: Tópico em Aprendizado de Máquina
Objetivos :

- Escolha dois datasets específicos para a tarefa de Regras de Associação. Não tente "produzir" um dataset como esse. Possivelmente, você irá falhar nisso. O dataset para Regras de Associação deve conter transações.
- Para cada dataset, você irá aplicar os algoritmos APriori e FP-Growth. Seu objetivo é minerar regras que contenham informações relevantes no dataset, seja porque alguma combinação de itens aparece com muita frequência, seja porque alguma combinação de itens não aparece com muita frequência mas está se destacando. Para isso, você deve variar os parâmetros de suporte, confiança e lift.
- Existem duas métricas não estudadas em sala de aula, 'leverage' e 'conviction', que estão disponibilizadas no pacote mlxtend para regras de associação. Estude essas métricas e explique como elas podem contribuir para as análises dos seus datasets. (Veja: http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/)
- No relatório, você deverá explicar como usou os parâmetros para a mineração e o que foi obtido. As regras selecionadas devem ser exibidas com suas medidas equivalentes (suporte, confiança, lift, leverage, conviction). Mostre as regras que você considerou mais relevantes e justifique por quê.
- Compare os resultados gerados pelos dois algoritmos. Conclua sobre as diferenças encontradas nos resultados de cada dataset na aplicação dos dois algoritmos.

1.1 0.1 Dependências

Para realização da tarefa foram utilizados as seguintes bibliotecas:

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
```

```
import seaborn as sns
import pandas_profiling

# Encoder
from mlxtend.preprocessing import TransactionEncoder

# Algoritmos
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import pyfpgrowth

#Metricas
from mlxtend.frequent_patterns import association_rules

import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

2 1. Dados

Este é um conjunto de dados de E-Commerce de Roupas Femininas que gira em torno das avaliações escritas pelos clientes

2.1 1.1 Informações sobre os dados:

Atributos:

- Clothing ID: Integer Variável categórica que se refere à peça específica que está sendo revisada.
- Age: Positiva Variável inteira da idade dos revisores.
- Title: variável de string para o título da revisão.
- Review Text: variável de string para o corpo de revisão.
- Rating: Variável Integral Positiva Positiva para a pontuação do produto concedida pelo cliente de 1 pior, a 5 melhores.
- Recommended IND: Variável binária indicando onde o cliente recomenda o produto, em que 1 é recomendado, 0 não é recomendado.
- Positive Feedback Count: Integral Positivo documentando o número de outros clientes que consideraram este comentário positivo.
- Division Name: Nome categórico da divisão de alto nível do produto.
- Department Name: nome categórico do nome do departamento do produto.
- Class Name: nome categórico do nome da classe do produto.

2.2 Importando Dataset

```
In [2]: roupas = pd.read_csv('Womens Clothing E-Commerce Reviews.csv')

In [3]: roupas.head()
```

```
Out[3]:    Unnamed: 0  Clothing ID  Age                    Title  \
        0           0          767   33                      NaN
        1           1         1080   34                      NaN
        2           2         1077   60  Some major design flaws
        3           3         1049   50         My favorite buy!
        4           4          847   47         Flattering shirt

                                           Review Text  Rating  Recommended IND  \
        0  Absolutely wonderful - silky and sexy and comf...       4                1
        1  Love this dress!  it's sooo pretty.  i happene...       5                1
        2  I had such high hopes for this dress and reall...       3                0
        3  I love, love, love this jumpsuit. it's fun, fl...       5                1
        4  This shirt is very flattering to all due to th...       5                1

           Positive Feedback Count   Division Name Department Name Class Name
        0                        0       Initmates        Intimate  Intimates
        1                        4         General         Dresses    Dresses
        2                        0         General         Dresses    Dresses
        3                        0  General Petite         Bottoms      Pants
        4                        6         General            Tops    Blouses

In [4]: roupas.drop(columns=['Unnamed: 0', 'Clothing ID', 'Review Text', 'Review Text', 'Title

In [5]: roupas.isna().sum()

Out[5]: Age                 0
        Rating              0
        Recommended IND     0
        Division Name      14
        Department Name    14
        Class Name         14
        dtype: int64

In [6]: roupas.dropna(inplace=True)
```

2.3 Discretizando

```
In [7]: roupas['Age'] = ['adult' if x < 60 else 'elderly' for x in roupas['Age']]

In [8]: roupas['Recommended IND'] = ['yes' if x == 1 else 'no' for x in roupas['Recommended IN

In [9]: list_aux = []
        for element in roupas['Rating']:
            if element == 5:
                list_aux.append('very good')
            elif element == 4:
                list_aux.append('good')
            elif element == 3:
                list_aux.append('normal')
```

```
            elif element == 2:
                list_aux.append('bad')
            else:
                list_aux.append('very bad')

        roupas['Rating'] = list_aux
```

```
In [10]: roupas_np = roupas.to_numpy()
```

```
In [17]: for coluna in roupas.columns:
             print(roupas[coluna].unique(), '\n')
```

```
['adult' 'elderly']

['good' 'very good' 'normal' 'bad' 'very bad']

['yes' 'no']

['Initmates' 'General' 'General Petite']

['Intimate' 'Dresses' 'Bottoms' 'Tops' 'Jackets' 'Trend']

['Intimates' 'Dresses' 'Pants' 'Blouses' 'Knits' 'Outerwear' 'Lounge'
 'Sweaters' 'Skirts' 'Fine gauge' 'Sleep' 'Jackets' 'Swim' 'Trend' 'Jeans'
 'Legwear' 'Shorts' 'Layering' 'Casual bottoms' 'Chemises']
```

```
In [16]: roupas.head()
```

Out[16]:

	Age	Rating	Recommended IND	Division Name	Department Name	Class Name
0	adult	good	yes	Initmates	Intimate	Intimates
1	adult	very good	yes	General	Dresses	Dresses
2	elderly	normal	no	General	Dresses	Dresses
3	adult	very good	yes	General Petite	Bottoms	Pants
4	adult	very good	yes	General	Tops	Blouses

2.4 Algoritmos

2.4.1 Apriori

```
In [11]: encoder = TransactionEncoder()
         roupas_encoded = encoder.fit(roupas_np).transform(roupas_np)
         roupas_encoded = pd.DataFrame(roupas_encoded, columns=encoder.columns_)
```

```
In [20]: frequent_itemsets = apriori(roupas_encoded, min_support=0.2, use_colnames=True)
         association_rules(frequent_itemsets, metric="confidence", min_threshold=0.8).sort_valu
```

Out[20]:

	antecedents	consequents	antecedent support	\
6	(Knits)	(Tops)	0.206331	
34	(adult, Tops, very good)	(yes)	0.207268	
26	(adult, very good)	(yes)	0.486409	
25	(Tops, very good)	(yes)	0.244163	
31	(General, adult, very good)	(yes)	0.280675	
13	(very good)	(yes)	0.558836	
19	(General, very good)	(yes)	0.324301	
12	(good)	(yes)	0.216300	
0	(Dresses)	(adult)	0.269214	
4	(General Petite)	(adult)	0.345944	
21	(yes, General Petite)	(adult)	0.285745	
2	(General)	(adult)	0.590065	
11	(yes)	(adult)	0.822256	
18	(General, yes)	(adult)	0.481979	
27	(yes, very good)	(adult)	0.557771	
9	(very good)	(adult)	0.558836	
28	(very good)	(adult, yes)	0.558836	
32	(General, yes, very good)	(adult)	0.323577	
16	(General, very good)	(adult)	0.324301	
7	(Tops)	(adult)	0.445978	
14	(General, Tops)	(adult)	0.291283	
33	(General, very good)	(adult, yes)	0.324301	
24	(Tops, yes)	(adult)	0.363540	
30	(General, Tops, yes)	(adult)	0.235685	
35	(Tops, yes, very good)	(adult)	0.243737	
22	(Tops, very good)	(adult)	0.244163	
36	(Tops, very good)	(adult, yes)	0.244163	
5	(General Petite)	(yes)	0.345944	
20	(adult, General Petite)	(yes)	0.305854	
10	(adult)	(yes)	0.881646	
3	(General)	(yes)	0.590065	
8	(Tops)	(yes)	0.445978	
17	(adult, General)	(yes)	0.518234	
15	(General, Tops)	(yes)	0.291283	
23	(adult, Tops)	(yes)	0.385651	
1	(Dresses)	(yes)	0.269214	
29	(adult, Tops, General)	(yes)	0.251747	

	consequent support	support	confidence	lift	leverage	conviction
6	0.445978	0.206331	1.000000	2.242262	0.114312	inf
34	0.822256	0.207013	0.998767	1.214666	0.036585	144.120512
26	0.822256	0.485685	0.998511	1.214355	0.085732	119.370574
25	0.822256	0.243737	0.998255	1.214044	0.042972	101.864911
31	0.822256	0.280164	0.998179	1.213951	0.049377	97.581288

13	0.822256	0.557771	0.998094	1.213848	0.098264	93.258562
19	0.822256	0.323577	0.997767	1.213450	0.056918	79.587353
12	0.822256	0.209143	0.966910	1.175922	0.031289	5.371457
0	0.881646	0.242757	0.901725	1.022774	0.005405	1.204312
4	0.881646	0.305854	0.884113	1.002798	0.000853	1.021289
21	0.881646	0.251278	0.879380	0.997429	-0.000648	0.981210
2	0.881646	0.518234	0.878267	0.996167	-0.001994	0.972242
11	0.881646	0.721413	0.877358	0.995136	-0.003526	0.965031
18	0.881646	0.421097	0.873685	0.990970	-0.003837	0.936974
27	0.881646	0.485685	0.870761	0.987653	-0.006072	0.915773
9	0.881646	0.486409	0.870397	0.987241	-0.006286	0.913204
28	0.721413	0.485685	0.869101	1.204721	0.082534	2.128264
32	0.881646	0.280164	0.865833	0.982064	-0.005117	0.882136
16	0.881646	0.280675	0.865476	0.981659	-0.005244	0.879794
7	0.881646	0.385651	0.864731	0.980814	-0.007544	0.874949
14	0.881646	0.251747	0.864268	0.980289	-0.005062	0.871966
33	0.721413	0.280164	0.863899	1.197510	0.046209	2.046917
24	0.881646	0.311861	0.857846	0.973005	-0.008652	0.832574
30	0.881646	0.201985	0.857014	0.972061	-0.005806	0.827728
35	0.881646	0.207013	0.849327	0.963342	-0.007877	0.785501
22	0.881646	0.207268	0.848892	0.962849	-0.007997	0.783240
36	0.721413	0.207013	0.847845	1.175257	0.030870	1.830944
5	0.822256	0.285745	0.825985	1.004535	0.001290	1.021429
20	0.822256	0.251278	0.821563	0.999157	-0.000212	0.996114
10	0.822256	0.721413	0.818256	0.995136	-0.003526	0.977992
3	0.822256	0.481979	0.816823	0.993392	-0.003206	0.970339
8	0.822256	0.363540	0.815151	0.991359	-0.003169	0.961561
17	0.822256	0.421097	0.812562	0.988210	-0.005024	0.948278
15	0.822256	0.235685	0.809127	0.984032	-0.003824	0.931214
23	0.822256	0.311861	0.808661	0.983466	-0.005243	0.928947
1	0.822256	0.217578	0.808197	0.982902	-0.003785	0.926702
29	0.822256	0.201985	0.802335	0.975773	-0.005015	0.899219

2.4.2 Fp-growth

```
In [13]: def sup_item(data, item):
             sup_item = 0
             for transacao in data:
                 if item in transacao:
                     sup_item += 1

             return sup_item
```

```
In [14]: def fp_growth(data, sup=10, conf=10):
             patterns = pyfpgrowth.find_frequent_patterns(data, sup*(len(data)))
             rules = pyfpgrowth.generate_association_rules(patterns, conf)
             list_aux = []
```

```
            for key, value in rules.items():

                try:
                    suport_x = patterns[key]/len(data)
                except:
                    suport_x = sup_item(data, key)/len(data)

                try:
                    suport_y = patterns[value[0]]/len(data)
                except:
                    suport_y = sup_item(data, value[0])/len(data)

                conf = value[1]
                suport_xy = conf*suport_x

                try:
                    conv = (1-suport_y)/(1-conf)

                except:
                    conv = float("inf")

                dict_aux = {'antecedents':key,
                            'consequents':value[0],
                            'antecedent support': suport_x,
                            'consequent support': suport_y,
                            'support': suport_xy,
                            'confidence': conf,
                            'lift': conf/suport_y,
                            'conviction': conv,
                            'leverage': (conf*suport_x)- suport_x*suport_y}
                list_aux.append(dict_aux)
            return pd.DataFrame(list_aux)[['antecedents','consequents','antecedent support','
                                           'support','confidence','lift','leverage','convictio
```

In [15]:
```
fp_growth(roupas_np, 0.2, 0.8)
```

Out[15]:

	antecedents	consequents	antecedent support	\
0	(Knits,)	(Tops,)	0.206331	
1	(good,)	(yes,)	0.216300	
2	(Dresses, General Petite)	(adult,)	0.220603	
3	(General Petite, adult)	(yes,)	0.305854	
4	(General Petite, yes)	(adult,)	0.285745	
5	(Tops, adult, very good)	(yes,)	0.207268	
6	(Tops, very good, yes)	(adult,)	0.243737	
7	(General, Tops, adult)	(yes,)	0.251747	
8	(General, Tops, yes)	(adult,)	0.235685	
9	(Tops, adult)	(yes,)	0.385651	
10	(Tops, yes)	(adult,)	0.363540	

11	(Dresses, adult, very good)	(yes,)	0.258436
12	(Dresses, very good, yes)	(adult,)	0.289025
13	(Dresses, General, adult)	(yes,)	0.285191
14	(Dresses, General, yes)	(adult,)	0.258095
15	(Dresses, adult)	(yes,)	0.485515
16	(Dresses, yes)	(adult,)	0.435157
17	(General, adult, very good)	(yes,)	0.280675
18	(General, very good, yes)	(adult,)	0.323577
19	(adult, very good)	(yes,)	0.486409
20	(very good, yes)	(adult,)	0.557771
21	(General, adult)	(yes,)	0.518234
22	(General, yes)	(adult,)	0.481979
23	(adult,)	(yes,)	0.881646
24	(yes,)	(adult,)	0.822256

	consequent support	support	confidence	lift	leverage	conviction
0	0.445978	0.206331	1.000000	2.242262	0.114312	inf
1	0.822256	0.209143	0.966910	1.175922	0.031289	5.371457
2	0.881646	0.200324	0.908073	1.029974	0.005830	1.287470
3	0.822256	0.251278	0.821563	0.999157	-0.000212	0.996114
4	0.881646	0.251278	0.879380	0.997429	-0.000648	0.981210
5	0.822256	0.207013	0.998767	1.214666	0.036585	144.120512
6	0.881646	0.207013	0.849327	0.963342	-0.007877	0.785501
7	0.822256	0.201985	0.802335	0.975773	-0.005015	0.899219
8	0.881646	0.201985	0.857014	0.972061	-0.005806	0.827728
9	0.822256	0.311861	0.808661	0.983466	-0.005243	0.928947
10	0.881646	0.311861	0.857846	0.973005	-0.008652	0.832574
11	0.822256	0.258010	0.998351	1.214161	0.045509	107.819325
12	0.881646	0.258010	0.892689	1.012525	0.003192	1.102901
13	0.822256	0.230999	0.809979	0.985069	-0.003501	0.935390
14	0.881646	0.230999	0.895015	1.015163	0.003450	1.127338
15	0.822256	0.391275	0.805897	0.980104	-0.007943	0.915718
16	0.881646	0.391275	0.899158	1.019863	0.007620	1.173656
17	0.822256	0.280164	0.998179	1.213951	0.049377	97.581288
18	0.881646	0.280164	0.865833	0.982064	-0.005117	0.882136
19	0.822256	0.485685	0.998511	1.214355	0.085732	119.370574
20	0.881646	0.485685	0.870761	0.987653	-0.006072	0.915773
21	0.822256	0.421097	0.812562	0.988210	-0.005024	0.948278
22	0.881646	0.421097	0.873685	0.990970	-0.003837	0.936974
23	0.822256	0.721413	0.818256	0.995136	-0.003526	0.977992
24	0.881646	0.721413	0.877358	0.995136	-0.003526	0.965031